基于 GPU 异构平台的实时 CT 图像重建系统的研究*

夏松竹[1†]，杨 静[1]，方宝辉[1]，徐金秀[2]

(1. 哈尔滨工程大学 计算机科学与技术学院, 哈尔滨 150000; 2. 江南计算技术研究所, 江苏 无锡 214083)

摘 要：针对采用单 CPU CT 图像重建时间长，采用 CPU 集群重建成本及能耗高的问题，提出了 CPU 多线程+GPU 的异构重建模型。这种模型采用 CPU 多线程流水线模式，将整个任务分解为若干个处理阶段，相邻的两个阶段之间以循环缓存连接，上一阶段完成一次计算任务后将数据放到循环缓存里，然后继续下一次的计算任务，下一阶段探测到循环缓存里有数据后，从缓存里取出数据开始计算。各个任务是并行处理任务的，针对某一耗时瓶颈模块再采用 GPU 并行加速，充分发挥 CPU 和 GPU 的计算资源。CPU 多线程+GPU 模型相对于 CPU 多线程模型加速 16.45 倍，相对于串行 CT 图像重建加速 20.5 倍以上。将 CPU 多线程+GPU 模型重建的图像与 CPU 串行程序重建的 CT 图像相比较，数据结果在误差范围内，满足实验设计要求。提出的图像重建模型采用成本较低的 GPU 显卡就实现了性能大幅提升，大大降低了 CT 图像重建系统的成本及功耗，而成本及功耗的降低会引起 CT 医疗诊断费用的降低，最终惠及广大病患。

关键词：GPU；CT 图像重建；流水线

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2017.12.0859

Research on real-time computed tomographic reconstruction system based on GPU heterogeneous platform

Xia Songzhu[1†], Yang Jing[1], Fang Baohui[1], Xu Jinxiu[2]

(1. *College of Computer Science & Technology*, *Harbin Engineering University*, *Harbin* 150000, *China*; 2. *Jiangnan Institute of Computing Technology*, *Wuxi Jiangsu* 214083, *China*)

Abstract: In order to solve long time consuming problems of single CPU computed tomographic reconstruction and the problems of high cost and high energy consumption in the reconstruction using CPU cluster, this paper proposed a heterogeneous reconstruction model of CPU multithreads +GPU . This model used the CPU multithread pipelining pattern, it divided the whole task into several stages, and it connected each two adjacent phases by a loop buffer. Once the calculation of the current task stage completes, the data would be put into the loop buffer, and then the next computing task would continue to execute. When the data in the loop buffers was detected by the latter stage, the data would be removed from the loop buffers and calculated. In this way, each thread processes tasks in a parallel way. For a time consuming bottleneck module, the parallel acceleration of GPU was adopted to give full play to the computing resources of CPU and GPU. The CPU multithreads +GPU model is 16.45 times faster than the CPU multithreaded model, and accelerates more than 20.5 times faster than serial CT image reconstruction. By comparing with experimental results of CPU serial program, the result of the CPU multithreads +GPU model can meet the experimental design requirement in the range of error tolerance. The image reconstruction model proposed by using low cost GPU has greatly enhanced performance, greatly reduces the cost and power consumption of the CT system . The reduction in cost and power consumption will lead to a reduction in the cost of CT medical diagnosis, which will eventually benefit patients.

Key words: GPU; CT image reconstruction; pipeline

收稿日期：2017-12-12；**修回日期：**2018-02-27　**基金项目：**国家重点研发计划资助项目（2017YFB0202702）；国家“973”计划资助项目（2014CB744100）；国家“863”计划资助项目（2012AA01A306）

作者简介：夏松竹（1973-），男（通信作者），河北东光人，副研究员，硕导，主要研究方向为信息安全、数据库应用技术（ctpapermail@126.com）；杨静（1962-），女，黑龙江哈尔滨人，教授，博导，主要研究方向为数据挖掘、数据与知识工程；方宝辉（1988-），男，山东烟台人，硕士研究生，主要研究方向为高性能计算、数据与知识工程；徐金秀（1963-），女，江苏溧阳人，高级工程师，主要研究方向为并行算法与优化．．

0 引言

CT（computed tomography）是指计算机断层重建技术，现在已经是医学诊断不可或缺的一种手段[1]。CT 图像重建具有计算量大、计算密集、高可并行等特点，医疗诊断对 CT 图像的成像速度有较高的要求。现在加速 CT 图像重建的主要有两种方法，一是减少图像重建过程中的浮点计算量，二是并行处理图像重建过程。前者是通过巧妙设计优化现有算法；后者利用高性能计算平台，提升运算速度[2]。GPU 具有计算核心多、计算能力强，被越来越多的应用在图形处理之外的通用计算[3]。GPU 在通用计算的普及可以使个人计算机获得大规模集群才能提供的计算能力，而且 GPU 具有成本低、功耗小的优势，可以大幅减少 CT 成本。本文的整体软件框架采用多线程流水线模式，将整个任务分解为若干个处理阶段，各个任务是并行的，充分利用 CPU 资源，然后针对耗时长的瓶颈模块，再采用 GPU 进一步加速。

1 CT 原理

根据朗伯定律,一单色 X 线束通过一密度均匀原子序数均匀的小物体时，其能量因与物质的原子相互作用而减弱[4,5]，减弱的程度可用下式表达：

$$I = I_0 e^{-ud}$$

其中：I_0为入射的 X 射线强度；

I为穿过物体后的 X 射线强度；

u 为物质对该波长的线性衰减系数； d 为穿过均匀密度物体的路径长度；

e 为自然对数底。

当高度准直的 X 射线束环绕人体某一部位作断面扫描时，其中一部分光子被人体组织吸收，X 射线强度衰减，未被吸收的剩余光子穿透人体后，被检测器接收，然后经放大并转化为电子流，作为原始的模拟信号数据输入到计算机进行运算处理，重建生成图像，然后将重建图像传给显示设备显示，供医疗诊断用[1]。

反投影过程是将滤波后的投影数据累加到输出图像中，通过不同角度下多次的叠加来实现反投影[7]。反投影重建过程简介如下。

输入：

反投影重建输入参数如表 1 所示。

表 1 反投影重建输入参数

Prodata	滤波后的投影数据
Rs	射线源到旋转中心的距离
Rd	探测器到射线焦点的距离
wd	沿着 X 方向的探测器宽度
cen_det	探测器中心位置
FOV	感兴趣区域尺寸
x_0, y_0	感兴趣区域中心相对于旋转中心的位置偏移
w, h	重建图像的宽度和高度
V, C	投影 View 数，以及 View 中的通道数

输出：

重建图像，尺寸为 w*h 的数组，数组中各点的值代表了感兴趣区域内各点的 CT 值。实现反投影的过程其实是一个逆向的过程，首先是图像坐标下各个像素点与图像中心的相对坐标位置，如图 1 所示。需要注意的是，在图像坐标系中 Y 轴向下为正；可计算出图中点（x_p, y_p)在该坐标系下与图像中心的相对位置，单位为像素。

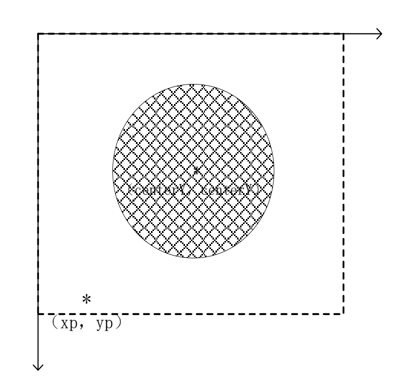


图 1 重建图像坐标系

此时的重建图像为投影坐标下的一个 FOV（图 2），可根据图像像素尺寸与 FOV 空间距离之间建立对应关系，将像素关系转为空间距离关系比例为 scale。

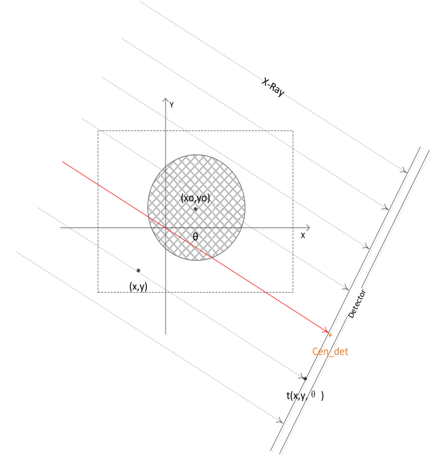


图 2 FOV 中心及平行投影示意图

在投影坐标下，经过旋转中心的射线在探测器上的投影位置为 cen_det。而 FOV 的中心与旋转中心存在位置的偏移，偏移量为（x_0, y_0)。根据像素-距离对应关系以及中心偏移量，可以通过以下公式计算出像素中的各点与旋转中心在空间的相对距离。

$$(x, y) = scale * [-(centerX, centerY)] + (x_0, y_0)$$

得到了（x，y）之后，可以通过以下公式计算出该点在探测器上的投影与旋转中心投影 cen_det 的相对距离。

$$t(x, y, \theta) = y\cos\theta - x\sin\theta$$

由于在探测器上的投影位置$cen_det + t(x, y, \theta)$未必是整数，需要根据其位置进行线性插值得到投影值；然后将投影值累加到图像中对应的像素，并乘上系数 π/V；最后将投影值

转换为 CT 值，单位为 HU。

$$CT = 1000 * \frac{\mu - \mu_{H2O}}{\mu_{H2O}}$$

2　实时 CT 系统实现

2.1　CT 系统架构

如图 3 所示，CT 系统的软件主要由三大部分组成，分别是扫描控制系统、用户接口、图像重建系统。扫描控制系统主要负责扫描床、机架的运动控制，用户接口是 CT 系统与用户交互的接口，图像重建系统负责对探测器采集的数据进行图像的重建。图像重建系统是本文的关注点，扫描控制系统和用户接口不作详述。

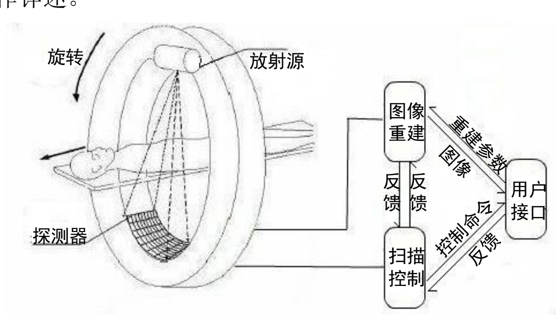


图 3　CT 系统

2.2　流水线图像重建系统

本文设计的 CT 系统图像重建系统采用的是 CPU 多线程加图像处理器（GPU）的异构混合加速模型。多线程采用的是多线程设计模式中的流水线模式，首先将 CT 图像重建任务划分为若干个处理阶段，上一个阶段任务的运算结果交给下一个阶段来继续处理，这样线程与线程之间是并行处理的，可以充分利用 CPU 的计算资源，进而提高整个图像重建系统的计算效率[1]。如图 4 所示，本文图像重建系统包括原始数据读取（RawData）、校准(correction)、扇形束转为平行束(rebin)、滤波(filter)、反投影重建(BPR)、图像后处理(img)等模块。每个模块是一个线程，上一个模块的计算结果放到循环 buffer 中，下一个模块检测到循环 buffer 中有数据，则从 buffer 中读出数据进行运算。图 5 是图像重建流水线框图。同一行中的各个模块表示不同次图像重建的相同处理阶段，相同颜色表示同一条流水线的不同处理阶段，如后缀为数字标记为 1 淡蓝色模块 RawData1、correction1、rebin1、filter1、BPR1、img1 分别表示第一条流水线的读取、校准、扇形束转为平行束、滤波、反投影重建、图像后处理等阶段。

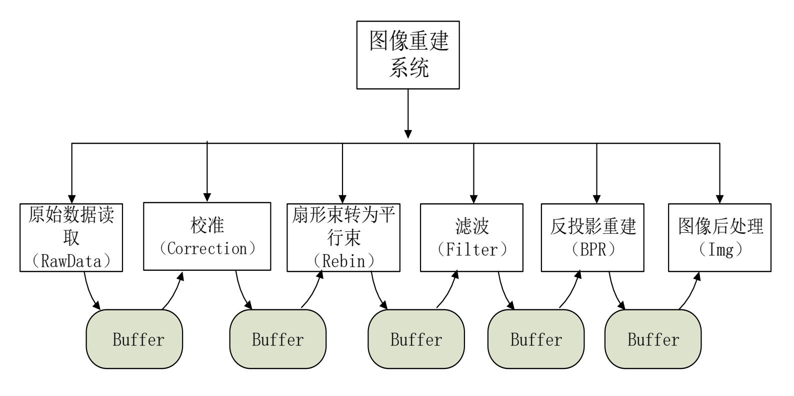


图 4　图像重建系统

Time	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11
thread1	RawData1	RawData2	RawData3	RawData4	RawData5	RawData6					
thread2		Correction1	Correction2	Correction3	Correction4	Correction5	Correction6				
thread3			Rebin1	Rebin2	Rebin3	Rebin4	Rebin5	Rebin6			
thread4				Filter1	Filter2	Filter3	Filter4	Filter5	Filter6		
thread5					BPR1	BPR2	BPR3	BPR4	BPR5	BPR6	
thread6						Img1	Img2	Img3	Img4	Img5	Img6

图 5　图像重建流水线框图

2.3　CUDA 架构

CUDA(compute unified device architecture)是 Nvidia 公司推出的专门应用于图像处理单元（GPU）进行通用计算的并行编程模型。简单地讲，CUDA 是便于编程人员利用 Nvidia GPU 进行通用计算的并行开发环境，目前支持 C/C++语言。在 CUDA 编程模型中，一个程序被分为 host 和 device 端。host 端是在 CPU 上运行的，device 端则是在被视为协处理器的 GPU 上运行[6]。由于 GPU 能够提供成百上千的计算核心，所以拥有极大的并行计算能力。

2.4　GPU 并行算法设计

通过软件耗时测试，rebin（扇形束转平行束）、filter（滤波）、BPR（反投影重建）这三个模块是整个图像重建软件的耗时瓶颈。因此，本文将这三个模块采用 GPU 进行 CUDA 并行，减少这三个模块的耗时，进而提升整个图像重建系统软件的运算速度。本文限于篇幅的原因，仅介绍最为核心反投影重建算法（BRP）GPU 并行设计。反投影重建算法的本质是对经过断层平面的某像素点的所有射线投影值求和。整幅图像的重建是将所有角度下投影值累加[8~10]。如下文反投影重建伪代

码，2～6 行计算探测器上各个像素点在某一旋转角度下的投影值；第 1 行进行调整旋转角度，计算该角度下各个像素点的投影值，然后累加到之前计算得到的相同像素点的投影值上。重复上述过程，直至所有旋转角度的投影值都累加完毕。本文图像域是 512*512 个像素点。因此每幅图像的重建需要作 views*512*512 次乘加操作。

反投影重建算法的伪代码：

```
1  for each view in sonogram domain
2   for each row in image domain
3     for each pixel of row
4        Accumulate the projection value through this pixel
5     end For
6  end For
7 end For
```

反投影重建并行设计方案：

比较直观的任务划分方式是在投影域按 view 进行划分，首先为每个线程开辟一个像素矩阵空间，每个线程负责几 view 像素点的累加，再将各个线程像素矩阵按对应像素进行规约求和。这种方法存在线程之间有相关性，需要各个线程的同步操作以及规约时的锁操作，耗时相对较多。本文是在图像域进行任务划分（图 6），1 个网格 Grid 里开启 6 个线程块（Block），每个 Block 里面有 512 个线程，每个 Block 负责图像坐标系下 85 或 86 行数据的重建，每个线程负责图像域每行的一个像素点的重建；优势是线程间没有关联性，减少同步带来的开销，减少 GPU 存储空间的开辟。

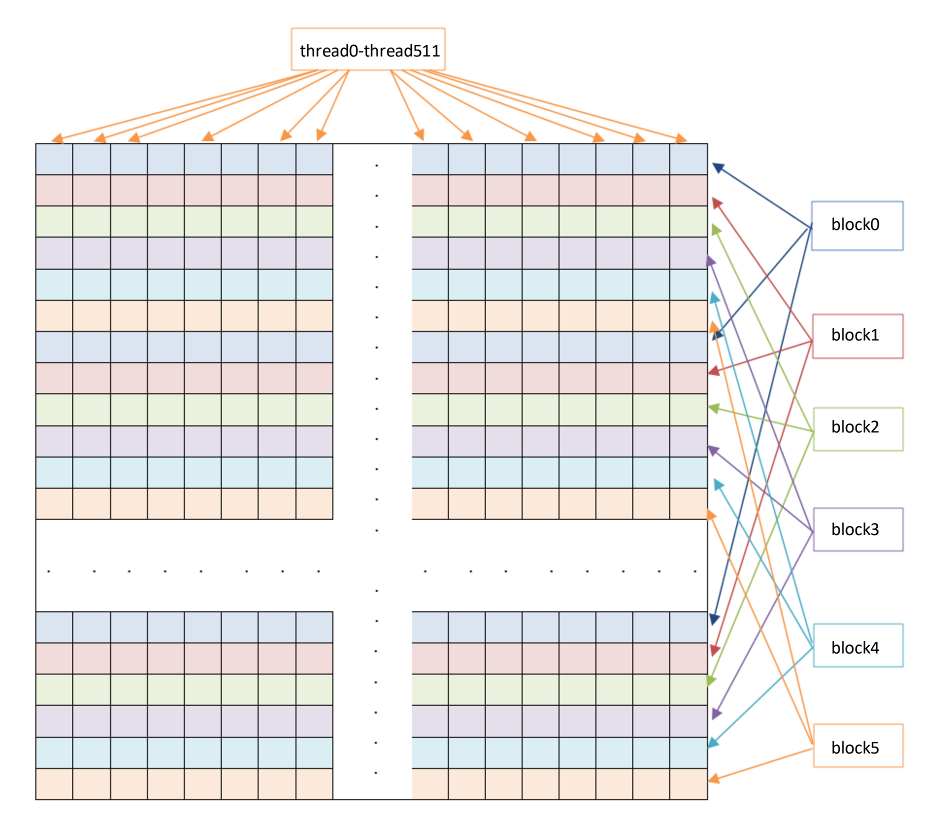


图 6　　并行任务划分

2.5　优化方案

a）采用异步流并行。Rebin、filter、BPR 这三个模块分别位于三个 cuda 流上，可以实现流之间数据拷贝与内核（kernel）计算的重叠[12～15]。

b)本文提出的流水线图像重建模型的各个模块之间是通过循环缓存连接。为了避免因为缓存过小造成某些节点运算堵塞和缓存过大造成存储资源浪费，本文通过反复实验来确定最佳的缓存大小。同时平衡各个节点的计算量，确保在优化后各个节点运算时间相当，避免某些节点过慢，成为整个运算流程的瓶颈，造成其前面的节点由于输出循环缓存满了而停滞工作，其后节点因输入循环缓存没有数据而停滞工作，进而影响到整个流水线的运算速度。

c）各个节点之间通过缓存连接，上一阶段完成一次计算任务后将数据放到循环缓存里，然后继续下一次的计算任务。下一阶段探测到循环缓存里有数据后，从缓存里取出数据开始计算，如果探测循环缓存里没有数据，则需要挂起线程，避免持续判断，造成 CPU 资源的极大浪费，需要适当调节线程的挂起时间，避免时间过长，造成本阶段任务的工时过长，影响整个任务的时间。通过实际测试，本文将各个节点探测不到数据后的挂起时间设置在 200 us，使得整个运算流程的耗时最短。

d）GPU 没有复杂的分支预测能力，同一 warp 中的 thread 必须执行相同的指令，如果 warp 中的线程进入不同的分支，那么同一时刻除了正在执行的分支，其余分支都被阻塞了，十分影响性能。因此，本文通过边界扩充（padding）去除 for 循环里面的用于判断图像重建范围的 if 判断语句。

e）由于 Global memory 寻址时钟周期是 Share memory

的 20～30 倍，所以本文将 Global memory 中的数据先拷贝到 Shared memory 中[11]，然后再从 shared memory 中取数据进行运算。这样每个线程只需一次 Global memory 寻址，128 次 Shared memory 寻址。

f）Device 端初始化过程中开辟 static memory，避免重建过程中反复开辟。

g）适当增加 grid 中的线程数目来提高 GPU 的 Occupancy（占用率）。流水线中的各个节点对于 GPU 资源是共享关系，同时各个节点之间也是竞争关系。随着更多的节点将任务放到 GPU 上运算，GPU 的 Occupancy 将有很大提升。但由于各个节点分属不同的 CPU 线程，各个线程抢占式的将自己的任务 offload 到 GPU 上，而 GPU 上的寄存器、共享缓存等硬件资源有限，必将导致某些节点提交到 GPU 上的计算任务由于得不到硬件资源而等待，同时引起很大的调度开销。因此图像重建速度首先会随着更多的计算任务放到 GPU 上而有所提升，同时 GPU 的 Occupancy 也会大幅增大，但到达某个临界点后，随着资源竞争大增大图像重建速度不会再提升，反而有下降，而 GPU 的 Occupancy 继续增加。因此，本文不将 Occupancy 作为本文图像重建程序的唯一性能指标，盲目追求 Occupancy 的增大。而是尽量平衡 GPU 与 CPU 的计算量，确保两者同时保持较高的占有率，如采用 openmp 加速 CPU 端的计算瓶颈。

3　实验结果及分析

测试平台如表 2 所示。

表 2　测试平台

Host	Device
处理器：Intel® Xeon® CPU E5-2630 v4 @ 2.20 GHz	芯片厂商 NVIDIA
OS Type：Linux 64 位	显卡芯片 GeForce GTX 1050 Ti
核数：物理核 10，逻辑核 20	显存类型 GDDR5
	显存容量 4 096 MB 显存位宽 128 bit 流处理单元 768

采用上述实验平台进行扫描，扫描人体模型长度 365 mm，重建层厚 5 mm，重建间隔 1，得到 780 幅 512*512 像素的重建图像，如图 7、8 所示。从图中能清楚地辨识出人体器官，通过与串行程序结果数值相比，误差在可接受范围内。

CPU 性能测试结果如表 3、4 所示。



图 7　纵隔窗重建图像



图 8　肺窗重建图像

表 3　CPU 性能测试结果

CPU 多线程		
模块	运算次数	时间/s
Rebin	780	36.0774
Filter	780	47.1297
BPR	780	327.098
总耗时	329s	

表 4　GPU 性能测试结果

CPU 多线程+GPU		
模块	运算次数	时间/s
Rebin	780	18.3326
Filter	780	17.3781
BPR	780	13.6265
总耗时	20s	

由表中可以看出，CPU 多线程+GPU 比仅有 CPU 多线程，Rebin 模块加速 1.96 倍，filter 模块加速 2.71 倍，BPR 加速

24 倍，整个图像重建软件加速 16.45 倍，加速效果明显。本文加速效果是 CPU 多线程+GPU 异构模型与 CPU 多线程流水线模型耗时比较，而 CPU 多线程本身是一种并行结构，对软件有加速效果。如果图像重建系统软件是串行结构，那么软件整体耗时是各个模块耗时之和，以本文测试为例，串行图像重建耗时应该大于 Rebin+Filter+BRP 耗时，即 410.3051 s。因此 CPU 多线程+GPU 异构模型相对于串行模型加速超过 20.5 倍。另外需要说明的是，本文所得的加速比是在 CPU 多线程流水线模型已经做了较好的优化的基础上比较得出的。

4 结束语

本文研究 CPU 多线程+GPU 的混合异构 CT 图像重建实现方法，重建速度有明显提升。这种 CPU 多线程+GPU 异构软件模型能够同时发挥 CPU 和 GPU 的计算能力，相对于集群并行重建会大幅减少成本及功耗，而成本及功耗的降低将在市场竞争中表现为价格的优势，这种价格优势必将最终惠及广大病患。本文提出的 CPU 多线程流水线模式+GPU 的异构并行软件模型不依赖于特定的算法，因此对于其他应用的性能优化也有一定的借鉴经验。另外，该方法也有良好的硬件可扩展性，随着计算量的增加，可以扩展更多的 GPU 硬件资源，提高运算速率。

参考文献：

[1] 杨英，张学军. X 射线在医学诊断领域的应用机理 [C]// 首届 X 射线管学术会议论文集. 北京：北京真空电子技术研究所，2015: 21-23.

[2] Tang M, Zhao J Y, Tong R F, *et al.* GPU accelerated convex hull computation [J]. Computers & Graphics, 2012, 36 (5): 498-506.

[3] 丁科，谭营. GPU 通用计算及其在计算智能领域的应用 [J]. 智能系统学报，2015, 10 (1): 1-11.

[4] Hsieh J. Computed tomography: principles, design, artifacts, and recent advances [M]. 2nd. Washington USA: SPIE Press, 2009: 36-41.

[5] Yu T, Wang R. Scene parsing using graph matching on street-view data [J]. Computer Vision and Image Understanding. 2016, 145: 70-80.

[6] 孙万捷，高瞻，潘海燕，等. 动态任务分配 CUDA 线程束步进体绘制 [J]. 计算机辅助设计与图形学学报，2016, 28 (10): 1630-1638.

[7] Dzmitry S. Real-Time 3D cone beam reconstruction [C]// Proc of IEEE Nuclear Science Symposium Conference Record. 2004: 3648-3652.

[8] Zhao X. GPU-based 3D cone-beam CT image construction: application to micro CT [C]// Proc of IEEE Nuclear Science Symposium Conference Record. 2007: 3922-3925.

[9] Kachelrieß M. Hyperfast parallel-beam backprojection [C]// Proc of IEEE Nuclear Science Symposium Conference Record. 2006 [C] : 3111-3114.

[10] Knaup M, Kachelrieß M. Real-time cone-beam CT image reconstruction using a mercury's dual cell-based system (DCBS) and a sony's playstation 3 (PS3) cluster [C]// Proc of IEEE Nuclear Science Symposium Conference Record. 2007: 3926-3928.

[11] 孙涛，韩善清，汪家旺. PET//CT 成像原理、优势及临床应用 [J]. 中国医学物理学杂志，2010, 27 (1): 1581-1587.

[12] 柳有权，刘学慧，吴恩华. 基于 GPU 带有复杂边界的三维实时流体模拟 [J]. 软件学报，2006, 17 (3): 568-576.

[13] NVIDIA. NVIDIA CUDA compute unified device architecture programming guide. 1. 0 edn [R]. Santa Clara, CA: [s. n.] , 2007.

[14] Seherl H, Keek B, Kowarsehik M, *et al.* Fast GPU-based CT reconstruction using the common unified device architecture (CUDA) [C]// Proc of IEEE Nuclear Science Symposium Conference. Honolulu, Hl: IEEE Press, 2007: 4464-4466.

[15] Pratx G, Chinn G, 01eott D P, *et al.* Fast, accurate and shift-varying line projections for iterative reconstruction using the GPU [J]. IEEE Trans on Medical Imaging, 2009, 28: 435-445.

[16] Smith B D. Image reconstruction from cone-beam projection: necessary and sufficient condition and reconstruction methods [J]. IEEE Trans on Medical Imaging, 1985, 4 (1): 14-25.

[17] Muller K, Xu F. Practical considerations for GPU-accelerated CT [C]// Proc of IEEE International Symposium on Biomedical Imaging: Macro to Nano. 2006: 1184-1187.